

Final Project-Bookstore Database- Yon Garber

CSC-434

Project overview-

- Short explanation on DB
- Relational Scheme
- Constrains
- Triggers that I implemented- explanation only
- SQL statements
- Triggers- required 4
- Bonus- ColdFusion and 2 extra triggers to my DB

My project is building a database for an online bookstore which sends books to its customers around the US. It has information about books, authors, publishers, employees, customers, etc. It had warehouses and employees. And it has an online system for purchases, returns, discounts, taxes etc. Relational scheme below-

Customer(Customer_ID, First_name, Last_name, address, phone_number, username, password)

Book(ISBN, title, author_ID, publisher_ID, date_published, price, quantity, genre, warehouse_ID)

Warehouse(warehouse_ID, location, phone_number, number_of_employees)

Employee(Employee_ID, address, DOB, phone_number, warehouse_ID, salary)

Order(Order_ID, customer_ID, date, total_value, Employee_ID_handeling, Shipment_Tracking_number)

Shippment(Shipment_Tracking_number, cost, shipment_address)

Book_order(ISBN, Order_ID, quantity)

Author(Author_ID, First_name, Last_name, address, phone, DOB)

Publisher(Publisher_ID, Company_name, address)

Reviews(review_ID, ISBN, customer_ID, title, content, date)

| Book_Store | |
|--------------------------|--|
| Tables | |
| Author | |
| Book | |
| customer | |
| Employee | |
| insert_books_Log | |
| orders | |
| orders_details | |
| Publisher | |
| review | |
| Shipment | |
| Shipment_cost | |
| Warehouse | |
| warehouse_salary_summary | |

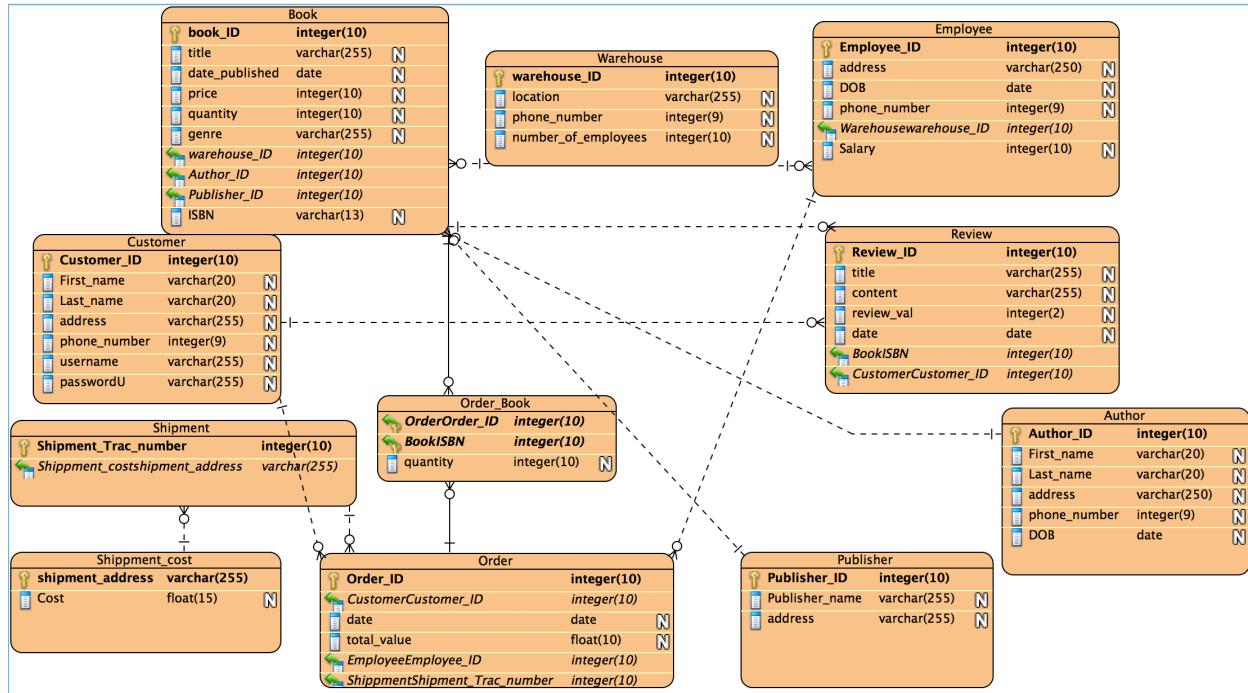
Constraints-

We have each book (for example Harry Potter is only in warehouse_id=1) in only one warehouse and not multiple warehouses.

Many domain attribute and referential integrity constraints were defined and can be seen in the ER diagram (below) and the relational scheme (above). Primary key (simple and composite) can be seen in the relational scheme and ER diagram as well.

Triggers that I implemented, and you can see them towards the end-
 Trigger to check if books available and if the order has quantity above zero. If not is stop the order and brings up an error message based on the mistake.
 Trigger that whenever an order is being made it decreases the quantity of books available in books.
 And the 4 required triggers from the assignment: referential integrity, domain attribute, summary, and log.

ER Model-



Relational schema with dependencies-

Customer(Customer_ID, First_name, Last_name, address, phone_number, username, password)

Cutomer_ID -> First_name

Cutomer_ID -> Last_name

Cutomer_ID -> address

Cutomer_ID -> phone_number

Cutomer_ID -> username

Cutomer_ID -> password

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Book(ISBN, title, author_ID, publisher_ID, date_published, price, quantity, genre, warehouse_ID)

ISBN -> title

ISBN -> author_ID
ISBN -> publisher_ID
ISBN -> date_published
ISBN -> price
ISBN -> quantity
ISBN -> genre
ISBN -> warehouse_ID

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Warehouse(warehouse_ID, location, phone_number, number_of_employees)

Warehouse_ID -> location
Warehouse_ID -> phone_number
Warehouse_ID -> number_of_employees

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Employee(Employee_ID, address, DOB, phone_number, warehouse_ID, salary)

Employee_ID -> address
Employee_ID -> DOB
Employee_ID -> phone_number
Employee_ID -> warehouse_ID
Employee_ID -> salary

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Order(Order_ID, customer_ID, date, total_value, Employee_ID_handeling,
Shipment_Tracking_number)

Order_ID -> customer_ID
Order_ID -> date
Order_ID -> total_value
Order_ID -> Employee_ID_handeling
Order_ID -> Shipment_Tracking_number

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Shippment(Shipment_Tracking_number, shipment_address)

Shipment_Tracking_number -> shipment_address

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Shipment_cost(shipment_address, cost)

shipment_address \rightarrow cost

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Book_order(ISBN, Order_ID, quantity)

ISBN, Order_ID \rightarrow quantity

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Author(Author_ID, First_name, Last_name, address, phone, DOB)

Author_ID \rightarrow First_name

Author_ID \rightarrow Last_name

Author_ID \rightarrow address

Author_ID \rightarrow phone

Author_ID \rightarrow DOB

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Publisher(Publisher_ID, Publisher_name, address)

Publisher_ID \rightarrow Publisher_name

Publisher_ID \rightarrow address

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

Reviews(review_ID, ISBN, customer_ID, title, content, date)

Review_ID \rightarrow ISBN

Review_ID \rightarrow customer_ID

Review_ID \rightarrow title

Review_ID \rightarrow content

Review_ID \rightarrow date

4NF- all values depend on the primary key with no transitive dependencies and no MVD.

SQL-

SELECT-

1.

```
select *  
from Employee  
where salary>400000;
```

| Result Grid | | | | | | | |
|-------------|------------|-----------|---------|------|--------------|--------------|---------|
| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 700000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1000000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 576500 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

2.

```
select AVG(salary),MIN(salary),MAX(salary),warehouse_id  
from employee  
group by warehouse_id;
```

| Result Grid | | | |
|-------------|-------------|-------------|--------------|
| AVG(salary) | MIN(salary) | MAX(salary) | warehouse_id |
| 553900.0000 | 143000 | 1000000 | 1 |
| 315750.0000 | 58000 | 900000 | 2 |

3.

```
select AVG(salary),warehouse_id  
from employee  
group by warehouse_id  
having avg(salary)>400000;
```

| AVG(salary) | warehouse_id |
|-------------|--------------|
| 553900.0000 | 1 |
| | |

4.

```
select *  
from employee
```

order by salary;

| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
|-------------|------------|-----------|---------|------|--------------|--------------|---------|
| 6 | Dalia | Fix | Lake St | NULL | 202345566 | 2 | 58000 |
| 9 | Jake | William | Pop St | NULL | 789310654 | 2 | 75000 |
| 4 | Mary | Zilber | Pop St | NULL | 789303654 | 1 | 143000 |
| 8 | Bear | Smith | Lake St | NULL | 202955559 | 2 | 230000 |
| 5 | Jane | Boading | Sun St | NULL | 123456789 | 1 | 350000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 576500 |
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 700000 |
| 7 | Inna | Long | Fox St | NULL | 202704322 | 2 | 900000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1000000 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

5.

```
select *  
from Employee  
where warehouse_id in (select warehouse_id from warehouse where warehouse_id=1);
```

| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
|-------------|------------|-----------|---------|------|--------------|--------------|---------|
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 700000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1000000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 576500 |
| 4 | Mary | Zilber | Pop St | NULL | 789303654 | 1 | 143000 |
| 5 | Jane | Boading | Sun St | NULL | 123456789 | 1 | 350000 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

6.

```
select employee_id,first_name, last_name, address from employee  
union  
select custoemr_id ,first_name,last_name, address from customer
```

| employee_id | first_name | last_name | address |
|-------------|------------|-----------|-------------|
| 1 | Yon | Garber | Door St |
| 2 | Rafi | Garber | Door St |
| 3 | Liat | Garber | Door St |
| 4 | Mary | Zilber | Pop St |
| 5 | Jane | Boading | Sun St |
| 6 | Dalia | Fix | Lake St |
| 7 | Inna | Long | Fox St |
| 8 | Bear | Smith | Lake St |
| 9 | Jake | William | Pop St |
| 4 | Rick | Jonson | Sea View St |
| 5 | Bob | Smith | Mass St |
| NULL | NULL | NULL | NULL |

INSERT-

1.

```
select *\nfrom Employee;
```

```
INSERT INTO Employee (employee_id,first_name, last_name, address,  
phone_number,warehouse_id, salary)  
VALUES (5, 'Jane', 'Boading', 'Sun St', 123456789, 1, 350000);
```

2.

```
select *\nfrom customer
```

```
insert into Customer(select employee_id,first_name, last_name, address, phone_number,  
warehouse_id, salary from employee where first_name='Yon');
```

| Result Grid | | | | | | |
|-------------|------------|-----------|---------|--------------|----------|-----------|
| custoemr_id | first_name | last_name | address | phone_number | username | passwordU |
| 1 | Yon | Garber | Door St | 202704369 | 1 | 700000 |
| HULL | NULL | NULL | NULL | NULL | NULL | NULL |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

3.

```
select *  
from customer
```

| Result Grid | | | | | | |
|-------------|------------|-----------|---------|--------------|----------|-----------|
| custoemr_id | first_name | last_name | address | phone_number | username | passwordU |
| 1 | Yon | Garber | Door St | 202704369 | 1 | 700000 |
| HULL | NULL | NULL | NULL | NULL | NULL | NULL |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

```
insert into Customer(select Employee.employee_id,Employee.first_name, Employee.last_name,  
Employee.address, Employee.phone_number, Warehouse.address,  
Warehouse.number_of_employees from Employee, Warehouse  
where Employee.warehouse_id=Warehouse.warehouse_id and first_name='Rafi');
```

| Result Grid | | | | | | |
|-------------|------------|-----------|---------|--------------|-------------|-----------|
| custoemr_id | first_name | last_name | address | phone_number | username | passwordU |
| 1 | Yon | Garber | Door St | 202704369 | 1 | 700000 |
| 2 | Rafi | Garber | Door St | 202704300 | Billbord St | 7 |
| HULL | NULL | NULL | NULL | NULL | NULL | NULL |
| | | | | | | |
| | | | | | | |
| | | | | | | |

DELETE-

1.

`select * from employee`

| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
|-------------|------------|-----------|---------|------|--------------|--------------|---------|
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 700000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1000000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 576500 |
| 4 | Mary | Zilber | Pop St | NULL | 789303654 | 1 | 143000 |
| 5 | Jane | Boading | Sun St | NULL | 123456789 | 1 | 350000 |
| 6 | Dalia | Fix | Lake St | NULL | 202345566 | 2 | 58000 |
| 7 | Inna | Long | Fox St | NULL | 202704322 | 2 | 900000 |
| 8 | Bear | Smith | Lake St | NULL | 202955559 | 2 | 230000 |
| 9 | Jake | William | Pop St | NULL | 789310654 | 2 | 75000 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

`delete from employee`

`where salary <=150000 and warehouse_id in
(select warehouse_id from warehouse where address='Billbord St');`

| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
|-------------|------------|-----------|---------|------|--------------|--------------|---------|
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 700000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1000000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 576500 |
| 5 | Jane | Boading | Sun St | NULL | 123456789 | 1 | 350000 |
| 6 | Dalia | Fix | Lake St | NULL | 202345566 | 2 | 58000 |
| 7 | Inna | Long | Fox St | NULL | 202704322 | 2 | 900000 |
| 8 | Bear | Smith | Lake St | NULL | 202955559 | 2 | 230000 |
| 9 | Jake | William | Pop St | NULL | 789310654 | 2 | 75000 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

(Mary Zilber was deleted).

2.

`select *
from customer;`

| custoemr_id | first_name | last_name | address | phone_number | username | password |
|-------------|------------|-----------|-------------|--------------|-------------|----------|
| 1 | Yon | Garber | Door St | 202704369 | 1 | 700000 |
| 2 | Rafi | Garber | Door St | 202704300 | Billbord St | 7 |
| 4 | Rick | Jonson | Sea View St | NULL | Ricktherock | 12345 |
| 5 | Bob | Smith | Mass St | NULL | BobSmith | 11111 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

`delete from customer
where phone_number=202704300;`

Rafi was deleted.

UPDATE-

1.

| Result Grid | | Filter Rows: | | Search | | Edit: | | Export/Import: | |
|-------------|------------|---------------|-------------|--------------|-------------|--------------|---|----------------|--|
| | | | | | | | | | |
| Customer ID | | Customer Name | | Address | | Phone Number | | User Details | |
| custoemr_id | first_name | last_name | address | phone_number | username | password | U | | |
| 1 | Yon | Garber | Door St | 202704369 | 1 | 700000 | | | |
| 4 | Rick | Jonson | Sea View St | NULL | Ricktherock | 12345 | | | |
| 5 | Bob | Smith | Mass St | NULL | BobSmith | 11111 | | | |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | | | |

```
update customer  
set username='USERnameYON'  
where first_name='Yon';
```

2.

```
select * from employee;
```

```

update employee
set salary=salary*1.1
where warehouse_id in (select warehouse_id from warehouse where
number_of_employees>5);

```

| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
|-------------|------------|-----------|---------|------|--------------|--------------|---------|
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 770000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1100000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 634150 |
| 5 | Jane | Boading | Sun St | NULL | 123456789 | 1 | 385000 |
| 6 | Dalia | Fix | Lake St | NULL | 202345566 | 2 | 58000 |
| 7 | Inna | Long | Fox St | NULL | 202704322 | 2 | 900000 |
| 8 | Bear | Smith | Lake St | NULL | 202955559 | 2 | 230000 |
| 9 | Jake | William | Pop St | NULL | 789310654 | 2 | 75000 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Salaries from warehouse_id= 1 increased by 10%.

CREATE VIEW-

1.

```

create view HighEarner as
select *
from employee
where salary>=300000;

```

```

select *
from HighEarner;

```

| Result Grid | | | | | | | |
|-------------|------------|-----------|---------|------|--------------|--------------|---------|
| | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 770000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1100000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 634150 |
| 5 | Jane | Boading | Sun St | NULL | 123456789 | 1 | 385000 |
| 7 | Inna | Long | Fox St | NULL | 202704322 | 2 | 900000 |

2.

```

Create view All_People as
select first_name, last_name from employee
union
select first_name, last_name from customer;

select *
from All_People;

```

| first_name | last_name |
|------------|-----------|
| Yon | Garber |
| Rafi | Garber |
| Liat | Garber |
| Jane | Boading |
| Dalia | Fix |
| Inna | Long |
| Bear | Smith |
| Jake | William |
| Rick | Jonson |
| Bob | Smith |

3.

```
select *
from HighEarner;
```

| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
|-------------|------------|-----------|---------|------|--------------|--------------|---------|
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 770000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1100000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 634150 |
| 5 | Jane | Boading | Sun St | NULL | 123456789 | 1 | 385000 |
| 7 | Inna | Long | Fox St | NULL | 202704322 | 2 | 900000 |

```
update HighEarner
set salary=200000
where first_name='Jane';
```

| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
|-------------|------------|-----------|---------|------|--------------|--------------|---------|
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 770000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1100000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 634150 |
| 7 | Inna | Long | Fox St | NULL | 202704322 | 2 | 900000 |

Because I updated Jane's salary to be lower than the threshold to become a high earner she is not in the HighEarner view anymore.

TRIGGERS-

1. Summary table for Employee- Gathering statistics-

```
create table warehouse_salary_summary(warehouse_id SERIAL PRIMARY KEY,min_salary
integer(20), max_salary integer(20), avg_salary integer(20));
```

Delimiter \$\$

```
create trigger employee_summary_insert after insert on Employee
for each row
begin
delete from warehouse_salary_summary;
insert warehouse_salary_summary
select warehouse_id, min(salary),max(salary),avg(salary) from employee group by
warehouse_id;
end; $$
```

```
INSERT INTO Employee (employee_id,first_name, last_name, address,
phone_number,warehouse_id, salary)
VALUE (10, 'Brad', 'Pit', 'LA', 555666777, 2, 5600000);
```

```
select *
from warehouse_salary_summary;
```

| Result Grid | | | |
|--------------|------------|------------|------------|
| warehouse_id | min_salary | max_salary | avg_salary |
| 1 | 200000 | 1100000 | 676038 |
| 2 | 58000 | 5600000 | 1372600 |
| NULL | NULL | NULL | NULL |

```
INSERT INTO Employee (employee_id,first_name, last_name, address,
phone_number,warehouse_id, salary)
VALUE(11, 'M&M', 'Rapper', 'Noname St', 999999999, 1, 7500000);
```

```
select *
from warehouse_salary_summary;
```

| Result Grid | | | |
|--------------|--------------|------------|------------|
| | Filter Rows: | Search | Edit: |
| warehouse_id | min_salary | max_salary | avg_salary |
| 1 | 200000 | 7500000 | 2040830 |
| 2 | 58000 | 5600000 | 1372600 |
| NULL | NULL | NULL | NULL |

2. Enforcing referential integrity-

Delimiter \$\$

```
create trigger warehouse_delete after delete on warehouse for each row
begin
delete from employee where warehouse_id=old.warehouse_id; end;
$$
```

```
INSERT INTO Warehouse (warehouse_id, address, phone_number,
number_of_employees)
VALUES (3, 'Delete St', 2027043720, 8);
```

```
INSERT INTO Employee (employee_id,first_name, last_name, address,
phone_number,warehouse_id, salary)
VALUES (12, 'Delete', 'Now', 'Big Lake St', 202345566, 3, 75000),
(13, 'David', 'Delete', 'Foxfox St', 202704322, 3, 340000),
(14, 'Dudu', 'Topaz', 'Poppop St', 789310654, 3, 24000);
```

```
select *
from employee;
```

| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
|-------------|------------|-----------|-------------|------|--------------|--------------|---------|
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 770000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1100000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 634150 |
| 5 | Jane | Boarding | Sun St | NULL | 123456789 | 1 | 200000 |
| 6 | Dalia | Fix | Lake St | NULL | 202345566 | 2 | 58000 |
| 7 | Inna | Long | Fox St | NULL | 202704322 | 2 | 900000 |
| 8 | Bear | Smith | Lake St | NULL | 202955559 | 2 | 230000 |
| 9 | Jake | William | Pop St | NULL | 789310654 | 2 | 75000 |
| 10 | Brad | Pit | LA | NULL | 555666777 | 2 | 5600000 |
| 11 | M&M | Rapper | Noname St | NULL | 999999999 | 1 | 7500000 |
| 12 | Delete | Now | Big Lake St | NULL | 202345566 | 3 | 75000 |
| 13 | David | Delete | Foxfox St | NULL | 202704322 | 3 | 340000 |
| 14 | Dudu | Topaz | Poppop St | NULL | 789310654 | 3 | 24000 |

```
select *
from warehouse;
```

| warehouse_id | address | phone_number | number_of_empl... |
|--------------|-------------|--------------|-------------------|
| 1 | Billbord St | 2027043720 | 7 |
| 3 | Delete St | 2027043720 | 8 |
| NULL | NULL | NULL | NULL |

```
delete from warehouse
where warehouse_id=3;
```

```
select *
from employee;
```

| employee_id | first_name | last_name | address | DOB | phone_number | warehouse_id | salary |
|-------------|------------|-----------|-----------|------|--------------|--------------|---------|
| 1 | Yon | Garber | Door St | NULL | 202704369 | 1 | 770000 |
| 2 | Rafi | Garber | Door St | NULL | 202704300 | 1 | 1100000 |
| 3 | Liat | Garber | Door St | NULL | 202978369 | 1 | 634150 |
| 5 | Jane | Boading | Sun St | NULL | 123456789 | 1 | 200000 |
| 6 | Dalia | Fix | Lake St | NULL | 202345566 | 2 | 58000 |
| 7 | Inna | Long | Fox St | NULL | 202704322 | 2 | 900000 |
| 8 | Bear | Smith | Lake St | NULL | 202955559 | 2 | 230000 |
| 9 | Jake | William | Pop St | NULL | 789310654 | 2 | 75000 |
| 10 | Brad | Pit | LA | NULL | 555666777 | 2 | 5600000 |
| 11 | M&M | Rapper | Noname St | NULL | 999999999 | 1 | 7500000 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

```
select *
from warehouse;
```

| warehouse_id | address | phone_number | number_of_empl... |
|--------------|-------------|--------------|-------------------|
| 1 | Billbord St | 2027043720 | 7 |
| NULL | NULL | NULL | NULL |

3. Trigger for Attribute domain checking-

Delimiter \$\$

Create trigger warehouse_id_domain_checking before insert on Employee

For each row

Begin

```
Declare temp int; set temp=0;
Select count(warehouse_id) into temp
From Warehouse where warehouse_id=new.warehouse_id;
If temp=0 then
    signal sqlstate '45000' set message_text = 'Warehouse_id DNE, wrong id'; End if;
End; $$
```

```
select *\nfrom employee;
```

```
select *\nfrom warehouse;
```

| warehouse_id | address | phone_number | number_of_empl... |
|--------------|-------------|--------------|-------------------|
| 1 | Billbord St | 2027043720 | 7 |
| 2 | Delete St | 2027043720 | 8 |
| NULL | NULL | NULL | NULL |

(warehouse_id= 3 DNE so it supposed to stop the insert and bring up a message)

```
INSERT INTO Employee (employee_id,first_name, last_name, address,  
phone_number,warehouse_id, salary)  
VALUES (12, 'Delete', 'Now', 'Big Lake St', 202345566, 3, 75000),  
(14, 'Dudu', 'Topaz', 'Popop St', 789310654, 3, 24000);
```

Message-

123 16:07:56 INSERT INTO Employee (employee_id,first_name, last_name, address, phone_numbr... Error Code: 1644. Warehouse_id DNE, wrong id

```
select *\nfrom employee;
```

4. Trigger for creating a DB log-

```
Create table insert_books_Log (message varchar(70));
```

```
select *  
from insert_books_Log;
```

| Result Grid | | Filter Rows: | Search | Export: |
|-------------|--|--------------|--------|---------|
| message | | | | |
| | | | | |
| | | | | |

```
select *  
from Book;
```

| ISBN | title | date_published | quantity | price | author_id | publisher_id |
|---------------|--------------------|----------------|----------|-------|-----------|--------------|
| 1234567891234 | Song of fire | 1909-09-14 | 50 | 125 | 1 | 1 |
| 2234567891234 | Harry Jogger | 2019-09-10 | 40 | 100 | 1 | 1 |
| 3234567891234 | Rock the Road | 2018-09-10 | 45 | 77 | 1 | 1 |
| 4234567891234 | Live long, dont... | 2000-01-10 | 45 | 23 | 2 | 2 |
| 5234567891234 | COVID-19 | 2020-01-01 | 30 | 39 | 2 | 2 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Delimiter \$\$

```
create trigger insert_book after insert on Book  
for each row  
begin  
insert into insert_books_Log values(concat('Book has been inserted by ',current_user(), ' at  
' ,CURRENT_TIMESTAMP()));  
end; $$
```

```
insert into Book (ISBN, title, date_published, quantity, price, author_id, publisher_id)  
values ('999967891234', 'SQL Insert', DATE '2020-04-03', 50, 200, 2,2),  
('223499991234', 'Harry Jogger is he Potter?', DATE '2019-05-25', 40, 75, 2,2);
```

```
select *  
from insert_books_Log;
```

message

Book has been inserted by root@localhost at 2020-04-04 16:29:53

Book has been inserted by root@localhost at 2020-04-04 16:29:53

```
select *  
from Book;
```

| ISBN | title | date_published | quantity | price | author_id | publisher_id |
|---------------|----------------------------|----------------|------------|------------|------------|--------------|
| 1234567891234 | Song of fire | 1909-09-14 | 50 | 125 | 1 | 1 |
| 2234567891234 | Harry Jogger | 2019-09-10 | 40 | 100 | 1 | 1 |
| 2234999991234 | Harry Jogger is he Potter? | 2019-05-25 | 40 | 75 | 2 | 2 |
| 3234567891234 | Rock the Road | 2018-09-10 | 45 | 77 | 1 | 1 |
| 4234567891234 | Live long, dont die | 2000-01-10 | 45 | 23 | 2 | 2 |
| 5234567891234 | COVID-19 | 2020-01-01 | 30 | 39 | 2 | 2 |
| 999967891234 | SQL Insert | 2020-04-03 | 50 | 200 | 2 | 2 |
| NUL | NUL | NUL | NUL | NUL | NUL | NUL |

BONUS POINTS- CF and extra triggers for my DB

ColdFusion Administrator <h1>Welcome to Yon's Site</h1> localhost:8500/Yon_website/SampleWebSite.htm

Welcome to Yon's Site



[Click here to see another picture of astonishing me](#)

View Customer List

[View Customer List](#)

Find Customer:

Customer's Name:

Add Customer:

Customer ID: Customer's First Name: Customer's Last Name: Customer's Address: Customer's Phone Number:
 Customer's Username: Customer's Password:

Delete Customer:

Customer's First Name:

Here I query a customer named yon.

ColdFusion Administrator Book Store DB localhost:8500/Yon_website/findcustomer.cfm

Find Customer's Record:

Customer ID: Customer's First Name: Customer's Last Name: Customer's Address: Customer's Phone Number:

[Back to Home](#)

Here is the info about him.

Here is a list of all customers in DB without their username and password (I don't want to show it to everyone):

ColdFusion Administrator View Customer Information localhost:8500/Yon_website/viewcustomer.cfm

CUSTOMER TABLE:

Customer ID : 1 Customer First Name : Yon Customer Last Name : Garber Customer Address : Door St Customer Phone Number : 202704369 Customer ID : 4 Customer First Name : Rick Customer Last Name : Jonson Customer Address : Sea View St Customer Phone Number : Customer ID : 5 Customer First Name : Bob Customer Last Name : Smith Customer Address : Mass St Customer Phone Number :

[Back to Home](#)

[Click here to see another picture of astonishing me](#)

View Customer List

[View Customer List](#)

Find Customer:

Customer's Name:

Add Customer:

Customer ID : 15 Customer's First Name: Johns Customer's Last Name: Hopkins Customer's Address: GW NW Customer's Phone Number: 2025064839
Customer's Username: Dontell Customer's Password: 121212

Delete Customer:

Customer's First Name:

Update Customer

Customer's First Name:

[Back to Home](#)

Here I add a customer to the bookstore DB.

Customer has been added

[Back to Home](#)

It was added successfully.

ColdFusion Administrator <h1>Welcome to Yon's Site </h1> +

localhost:8500/Yon_website/SampleWebSite.htm



[Click here to see another picture of astonishing me](#)

View Customer List

[View Customer List](#)

Find Customer:

Customer's Name:

Add Customer:

Customer ID: Customer's First Name: Customer's Last Name: Customer's Address: Customer's Phone Number:
 Customer's Username: Customer's Password:

Delete Customer:

Customer's First Name:

Update Customer

Customer's First Name:

[Back to Home](#)

Here I delete a customer from the DB: Johns

ColdFusion Administrator Delete Customer +

localhost:8500/Yon_website/deletecustomer.cfm

Customer has been deleted

[Back to Home](#)

It was deleted successfully.

Before-

| custoemr_id | first_name | last_name | address | phone_number | username | passwo |
|-------------|------------|-----------|-------------|--------------|-------------|--------|
| 1 | Yon | Garber | Door St | 202704369 | USERnameYON | 70000 |
| 4 | Rick | Jonson | Sea View St | NULL | Ricktherock | 12345 |
| 5 | Bob | Smith | Mass St | NULL | BobSmith | 11111 |
| 15 | Johns | Hopkins | American | 202436794 | Donttell | 1212 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

After-

| custoemr_id | first_name | last_name | address | phone_number | username | passwo |
|-------------|------------|-----------|-------------|--------------|-------------|--------|
| 1 | Yon | Garber | Door St | 202704369 | USERnameYON | 70000 |
| 4 | Rick | Jonson | Sea View St | NULL | Ricktherock | 12345 |
| 5 | Bob | Smith | Mass St | NULL | BobSmith | 11111 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Here I update Johns's record and also changed the color of the background:

ColdFusion Administrator Welcome to Yon's Site

localhost:8500/Yon_website/SampleWebSite.htm



[Click here to see another picture of astonishing me](#)

View Customer List

[View Customer List](#)

Find Customer:

Customer's Name:

Add Customer:

Customer ID : Customer's First Name: Customer's Last Name: Customer's Address: Customer's Phone Number:
 Customer's Username: Customer's Password:

Delete Customer:

Customer's First Name:

Update Customer

Customer's First Name:
[Back to Home](#)

From address of GW to American:

ColdFusion Administrator Book Store Database

localhost:8500/Yon_website/updatefindcustomer.cfm

Updating Customer's Record:

Customer ID : 15 Customer's First Name: Johns Customer's Last Name: Hopkins Customer's Address: GW Customer's Phone Number: 202436794
 Customer's Username: Dontell Customer's Password: 1212

ColdFusion Administrator Book Store Database

localhost:8500/Yon_website/updatefindcustomer.cfm

Updating Customer's Record:

Customer ID : 15 Customer's First Name: Johns Customer's Last Name: Hopkins Customer's Address: American Customer's Phone Number: 202436794
 Customer's Username: Dontell Customer's Password: 1212

And I clicked Update Customer's record which was successful.

ColdFusion Administrator Update customer Record

localhost:8500/Yon_website/updateupdatecustomer.cfm

Customer's Record has been updated

[Back to Home](#)

I query another customer:Rick

The screenshot shows a web browser window with the title "ColdFusion Administrator" and the URL "localhost:8500/Yon_website/SampleWebSite.htm". The page displays a photograph of a man in a suit holding a small orange box. Below the photo is a link: "Click here to see another picture of astonishing me". The main content area has a yellow background and contains several sections: "View Customer List" with a "View Customer List" link; "Find Customer:" with a search input field containing "Rick" and a "Find Customer" button; "Add Customer:" with input fields for Customer ID, First Name, Last Name, Address, and Phone Number; "Delete Customer:" with an input field for First Name and a "Delete Customer" button; and "Update Customer" with an input field for First Name and a "Update Customer" button. There is also a "Back to Home" link.

Find Customer's Record:

Customer ID : Customer's First Name: Customer's Last Name: Customer's Address: Customer's Phone Number:

[Back to Home](#)

And I query another customer named Bob:

ColdFusion Administrator Welcome to Yon's Site

localhost:8500/Yon_website/SampleWebSite.htm



[Click here to see another picture of astonishing me](#)

View Customer List

[View Customer List](#)

Find Customer:

Customer's Name:

Add Customer:

Customer ID : Customer's First Name: Customer's Last Name: Customer's Address: Customer's Phone Number:
 Customer's Username: Customer's Password:

Delete Customer:

Customer's First Name:

Update Customer

Customer's First Name:

[Back to Home](#)

And here is the result:

ColdFusion Administrator Book Store DB

localhost:8500/Yon_website/findcustomer.cfm

Find Customer's Record:

Customer ID : Customer's First Name: Customer's Last Name: Customer's Address: Customer's Phone Number:

[Back to Home](#)

1. Trigger to check if books available and if the order has quantity above zero. If not is stop the order and brings up an error message based on the mistake.

Delimiter \$\$

```
create trigger is_available before insert on orders_details
```

```
for each row
```

```
begin
```

```
IF new.quantity <= 0
```

```
THEN signal sqlstate '45000' set message_text = 'you should buy something';END IF;
```

```
IF new.quantity > (SELECT book.quantity
```

```
        FROM book
```

```
        WHERE new.book_id = book.ISBN
```

```
        LIMIT 1)
```

```
THEN
```

```
signal sqlstate '45000' set message_text = 'NOT AVAILABLE';
END IF;
END; $$
```

```
select *
from book;
```

| ISBN | title | date_published | quantity | price | author_id | publisher_id |
|----------------|--------------------|----------------|----------|-------|-----------|--------------|
| 12345678912... | Song of fire | 1909-09-14 | 50 | 125 | 1 | 1 |
| 22345678912... | Harry Jogger | 2019-09-10 | 40 | 100 | 1 | 1 |
| 22349999912... | Harry Jogger i... | 2019-05-25 | 40 | 75 | 2 | 2 |
| 32345678912... | Rock the Road | 2018-09-10 | 45 | 77 | 1 | 1 |
| 42345678912... | Live long, dont... | 2000-01-10 | 45 | 23 | 2 | 2 |
| 52345678912... | COVID-19 | 2020-01-01 | 30 | 39 | 2 | 2 |
| 999967891234 | SQL Insert | 2020-04-03 | 50 | 200 | 2 | 2 |
| HULL | HULL | HULL | HULL | HULL | HULL | HULL |

```
INSERT INTO orders_details (book_id, order_id ,quantity)
VALUES (2234999991234, 1,0),
(3234567891234, 2, 20);
```

✖ 43 15:21:23 INSERT INTO orders_details (book_id, order_id ,quantity) VALUES (2234999991234, 1,0),... Error Code: 1644. you should buy something

The first value in the insert is zero for the quantity.

```
INSERT INTO orders_details (book_id, order_id ,quantity)
VALUES (2234999991234, 1,5),
(3234567891234, 2, 1000);
```

✖ 47 15:22:57 INSERT INTO orders_details (book_id, order_id ,quantity) VALUES (2234999991234, 1,5),... Error Code: 1644. NOT AVAILABLE

The second value requires too many books that we don't have in stock.

```
INSERT INTO orders_details (book_id, order_id ,quantity)
VALUES (2234999991234, 1,5),
(3234567891234, 2, 4);
```

✓ 49 15:23:33 INSERT INTO orders_details (book_id, order_id ,quantity) VALUES (2234999991234, 1,5),... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0

```
select * from orders_details;
```

| Result Grid | | | | | Filter Rows: | Search | Export: |
|----------------|----------|----------|--|--|--------------|--------|---------|
| book_id | order_id | quantity | | | | | |
| 22349999912... | 1 | 5 | | | | | |
| 32345678912... | 2 | 4 | | | | | |

2. Trigger that whenever an order is being made it decreases the quantity of books available in books.

Delimiter \$\$

```
create trigger adjust_quant after insert on orders_details
for each row
begin
    if new.quantity>0 then
        update book set book.quantity=book.quantity-new.quantity where ISBN=new.book_id;
    END IF;
END; $$
```

delete from orders_details

(here I just make sure that orders_details is clear from previous steps)

```
select *
from book;
```

| Result Grid | | | | | | | | | | Filter Rows: | Search | Edit: | Export/Import: |
|----------------|--------------------|----------------|----------|-------|-----------|--------------|--|--|--|--------------|--------|-------|----------------|
| ISBN | title | date_published | quantity | price | author_id | publisher_id | | | | | | | |
| 12345678912... | Song of fire | 1909-09-14 | 50 | 125 | 1 | 1 | | | | | | | |
| 22345678912... | Harry Jogger | 2019-09-10 | 40 | 100 | 1 | 1 | | | | | | | |
| 22349999912... | Harry Jogger i... | 2019-05-25 | 40 | 75 | 2 | 2 | | | | | | | |
| 32345678912... | Rock the Road | 2018-09-10 | 45 | 77 | 1 | 1 | | | | | | | |
| 42345678912... | Live long, dont... | 2000-01-10 | 45 | 23 | 2 | 2 | | | | | | | |
| 52345678912... | COVID-19 | 2020-01-01 | 30 | 39 | 2 | 2 | | | | | | | |
| 999967891234 | SQL Insert | 2020-04-03 | 50 | 200 | 2 | 2 | | | | | | | |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | | | | | | | |

```
INSERT INTO orders_details (book_id, order_id ,quantity)
VALUES (2234999991234, 1,7),
(3234567891234, 2, 9);
```

```
select *
from book;
```

Result Grid | Filter Rows: Search | Edit: | Export/Import:

| ISBN | title | date_published | quantity | price | author_id | publisher_id |
|----------------|--------------------|----------------|----------|-------|-----------|--------------|
| 12345678912... | Song of fire | 1909-09-14 | 50 | 125 | 1 | 1 |
| 22345678912... | Harry Jogger | 2019-09-10 | 40 | 100 | 1 | 1 |
| 22349999912... | Harry Jogger i... | 2019-05-25 | 33 | 75 | 2 | 2 |
| 32345678912... | Rock the Road | 2018-09-10 | 36 | 77 | 1 | 1 |
| 42345678912... | Live long, dont... | 2000-01-10 | 45 | 23 | 2 | 2 |
| 52345678912... | COVID-19 | 2020-01-01 | 30 | 39 | 2 | 2 |
| 999967891234 | SQL Insert | 2020-04-03 | 50 | 200 | 2 | 2 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

select * from orders_details

Result Grid | Filter Rows: Search

| book_id | order_id | quantity |
|----------------|----------|----------|
| 22349999912... | 1 | 7 |
| 32345678912... | 2 | 9 |